Chapter 20. Zip Compression

The PDS standards support two different approaches to data compression:

1. In one case, a data object contains numbers that have been encoded using one of several supported methods (e.g., “Huffman first difference”). In this approach, the label describes the compressed data and the ENCODING_TYPE keyword indicates how the data object is to be decompressed by the user. PDS standards support this approach to compression for IMAGE objects only. For more information on compression of individual IMAGE objects, see Section A.19.

2. In the alternative approach, a standard compression method called “Zip” is used. In this case, an entire data file is compressed rather than a particular data object. The user is expected to apply an “Unzip” utility to decompress the file, and the label then describes the decompressed data directly.

This chapter describes PDS standards for archiving data using Zip compression. In general, the archiving of data in a compressed format should be used sparingly. Although compression reduces the number of physical volumes, it makes the data more difficult for users to interpret. PDS recommends that data compression be used only in limited situations, such as to compress very large and infrequently used data, or to archive processed data where the source product is readily available in a non-compressed PDS archive.

20.1 Zip Software

The Zip method was chosen because the algorithm and supporting software for all major platforms are available without charge to the general user community. The *Info-Zip Consortium* and Info-Zip working group, for example, provide information and software at this URL:

http://www.cdrom.com/pub/infozip

This same information is available on line from PDS at:

http://pds.jpl.nasa.gov

20.2 Zip File Labels

When archiving data in Zip format, two files need to be considered: (1) the zip file itself, and (2) the data file produced by decompressing the zip file. PDS strongly recommends that these two files have the same name but different extensions: “.ZIP” for the zip file and a more descriptive extension (e.g., “.DAT” or “.IMG”) for the unzipped file. The “.ZIP” file extension is reserved exclusively for zip-compressed files within the PDS.

PDS does not recommend the practice of compressing multiple data files into a single zip file, unless those files reside in the same directory and have the same name, but different extensions. For example, if file “ABC.IMG” contains an image and file “ABC.TAB” contains a table of additional information relevant to that image, then both files can be archived in the file “ABC.ZIP”. This will minimize the potential confusion for a user who may not be able to locate a desired file because it is hidden inside a zip file with a different name.

Like all PDS data files, both the zipped and the unzipped data files require labels. Both files must be described by a single, detached PDS label file using the combined-detached label approach (see Section 5.2.2). Attached labels are not permitted for Zip-compressed data, because the user must be able to examine the label before deciding whether or not to decompress the file. In a combined-detached label, each individual file is described within a FILE object. Here is the general framework:

```
PDS_VERSION_ID          = PDS3
DATA_SET_ID             = ...
PRODUCT_ID              = ...
   (other parameters relevant to both Zipped and Unzipped files)

OBJECT                  = COMPRESSED_FILE
   (parameters describing the compressed file)
END_OBJECT              = COMPRESSED_FILE

OBJECT                  = UNCOMPRESSED_FILE
   (parameters describing the first uncompressed file)
END_OBJECT              = UNCOMPRESSED_FILE

OBJECT                  = UNCOMPRESSED_FILE
   (parameters describing the a second uncompressed file, if present)
END_OBJECT              = UNCOMPRESSED_FILE
END
```

The first FILE object, the COMPRESSED_FILE, refers to the zipped file; additional FILE objects, called UNCOMPRESSED_FILEs, refer to the decompressed data file(s) that the user will obtain by unzipping the first.

The zip file is described via a “minimal label” (see Section 5.2.3). The following keywords are required:

```
FILE_NAME               = name of the zipfile
RECORD_TYPE             = UNDEFINED
ENCODING_TYPE           = ZIP
INTERCHANGE_FORMAT      = BINARY
UNCOMPRESSED_FILE_NAME  = a list of the names of all the files archived
                           in the zipfile
REQUIRED_STORAGE_BYTES  = approximate total number of bytes in the data
                           files
DESCRIPTION             = a brief description of the zipfile format
```

Typically, the DESCRIPTION is given as a pointer to a file called “ZIPINFO.TXT” found in the DOCUMENT directory on the same volume.

The subsequent UNCOMPRESSED_FILE object(s) contain complete descriptions of the data files obtained by unzipping the zip file.

20.3 Packaging Zip Archives on Volumes

A volume containing zip files with combined-detached labels as presented above conforms to all established PDS standards *provided both the zip file and its constituent data files are archived*. The unique feature of a Zip-compressed PDS archive volume is that only the zip files appear; the UNCOMPRESSED_FILE objects described by the labels are not present on the volume, but can be obtained by unzipping the zip files provided.

In the interests of long-term archiving, a PDS archive zip file must include all the support files required to completely reconstitute the labeled data files. Specifically, the zipped archive must include not only the data files, but also the label file(s) for the uncompressed data. Ideally, any .FMT files referenced by ^STRUCTURE keywords in the labels should also be included in the zip file.

Note: These additional .LBL and .FMT files do not need to be described by UNCOMPRESSED_FILE objects in the label, because PDS label and format files never require labels. Furthermore, the sizes of these files do not need to be included in the value of the REQUIRED_STORAGE_BYTES keyword. However, the names of these files do need to be included in the list of UNCOMPRESSED_FILE_NAME values.

20.4 Label Example

The following is an example of a PDS label for a Zip-compressed data file.

```
PDS_VERSION_ID             = PDS3
DATA_SET_ID                = "HST-S-WFPC2-4-RPX-V1.0"
SOURCE_FILE_NAME           = "U2ON0101T.SHF"
PRODUCT_TYPE               = OBSERVATION_HEADER
PRODUCT_CREATION_TIME      = 1998-01-31T12:00:00

OBJECT                     = COMPRESSED_FILE
  FILE_NAME                = "0101_SHF.ZIP"
  RECORD_TYPE              = UNDEFINED
  ENCODING_TYPE            = ZIP
  INTERCHANGE_FORMAT       = BINARY
  UNCOMPRESSED_FILE_NAME   = {"0101_SHF.DAT", "0101_SHF.LBL"}
  REQUIRED_STORAGE_BYTES   = 34560
  ^DESCRIPTION             = "ZIPINFO.TXT"
END_OBJECT                 = COMPRESSED_FILE

OBJECT                     = UNCOMPRESSED_FILE
  FILE_NAME                = "0101_SHF.DAT"
  RECORD_TYPE              = FIXED_LENGTH
  RECORD_BYTES             = 2880
  FILE_RECORDS             = 12
  ^FITS_HEADER             = ("0101_SHF.DAT",     1 <BYTES>)
  ^HEADER_TABLE            = ("0101_SHF.DAT", 25921 <BYTES>)

  OBJECT                   = FITS_HEADER
    HEADER_TYPE            = FITS
    INTERCHANGE_FORMAT    = ASCII
    RECORDS               = 7
    BYTES                 = 20160
    ^DESCRIPTION          = "FITS.TXT"
  END_OBJECT              = FITS_HEADER

  OBJECT                  = HEADER_TABLE
    NAME                  = HEADER_PACKET
    INTERCHANGE_FORMAT    = BINARY
    ROWS                  = 965
    COLUMNS               = 1

    ROW_BYTES             = 2
    DESCRIPTION           = "This is the HST standard header packet
                            containing observation parameters.  It is
                            stored as a sequence of 965 two-byte
                            integers.  For more detailed information,
                            contact Space Telescope Science Institute."

    OBJECT                = COLUMN
      NAME                = PACKET_VALUES
      DATA_TYPE           = MSB_INTEGER
      START_BYTE          = 1
      BYTES               = 2
    END_OBJECT            = COLUMN
  END_OBJECT              = HEADER_TABLE

END_OBJECT                = UNCOMPRESSED_FILE
END
```

20.5 ZIPINFO.TXT Example

While the ZIPINFO.TXT file is not required, it is strongly recommended that this file be included as part of the process of documenting the contents of a zip file. The following is an example ZIPINFO.TXT file and the type of information that should be included in the ZIPINFO.TXT file:

```
PDS_VERSION_ID            = PDS3
RECORD_TYPE               = STREAM

OBJECT                    = TEXT
  PUBLICATION_DATE        = 1999-07-26
  NOTE                    = "This file provides an overview of the ZIP
                            file format."

END_OBJECT                = TEXT
END

Many of the files in this data set are compressed using Zip format.
They are all indicated by the extension ".ZIP".  ZIP is a utility that
compresses files and also allows for multiple files to be stored in a
single Zip archive.  You will need the UNZIP utility to extract the
files.

The SOFTWARE directory on this volume contains a complete description of
the Zip file format and also the complete source code for the UNZIP
utility.  The file format and file decompression algorithms are
described in the file SOFTWARE/APPNOTE.TXT.
```

```
It is far simpler to obtain a pre-built binary of the UNZIP application
for your platform.  Binaries for most platforms are available from the
Info-ZIP web site, currently at:

        http://www.cdrom.com/pub/infozip/

The same information can also be found a the PDS Central Node's web
site, currently at:

        http://pds.jpl.nasa.gov/
```

20.6 Additional Files

As of this writing, Zip appears to be a robust standard with a long future of general use. Nevertheless, PDS long-term archiving goals reach well past the lifetime of many popular standards, past and present. For this reason, any volume containing zip files is required to contain a complete description of the zip file format with sample “Unzip” source code. This information must be located in an appropriate subdirectory of the SOFTWARE directory tree. The required text and source code may be obtained directly from the Info-Zip web site or by contacting a Central Node data engineer.